“Longhorn” Client Architecture

Michael Wallent
General Manager, Windows Client Platform
Microsoft Corporation

THIRD ANNUAL

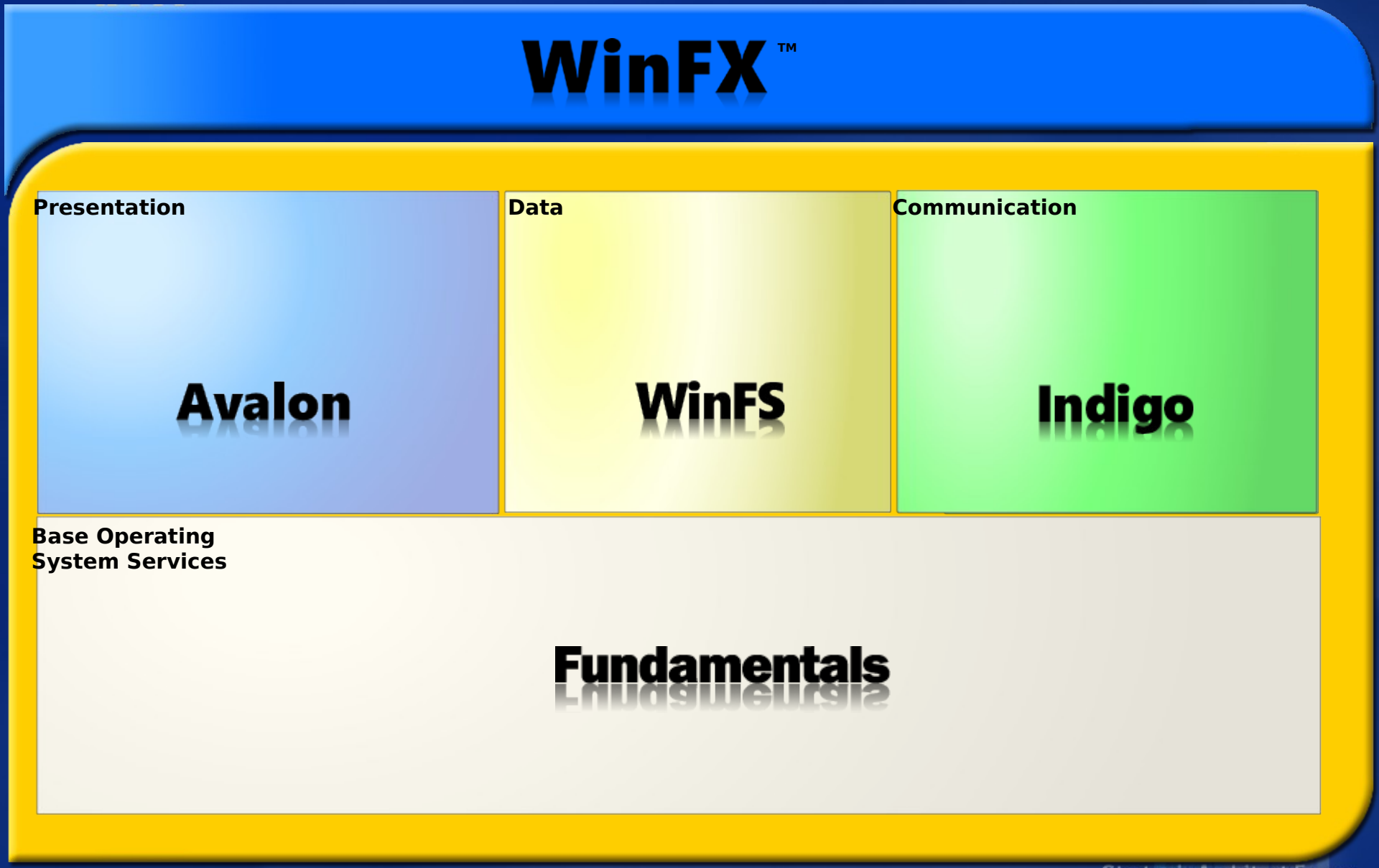
2003 Microsoft® **STRATEGIC ARCHITECT FORUM**

for Customers

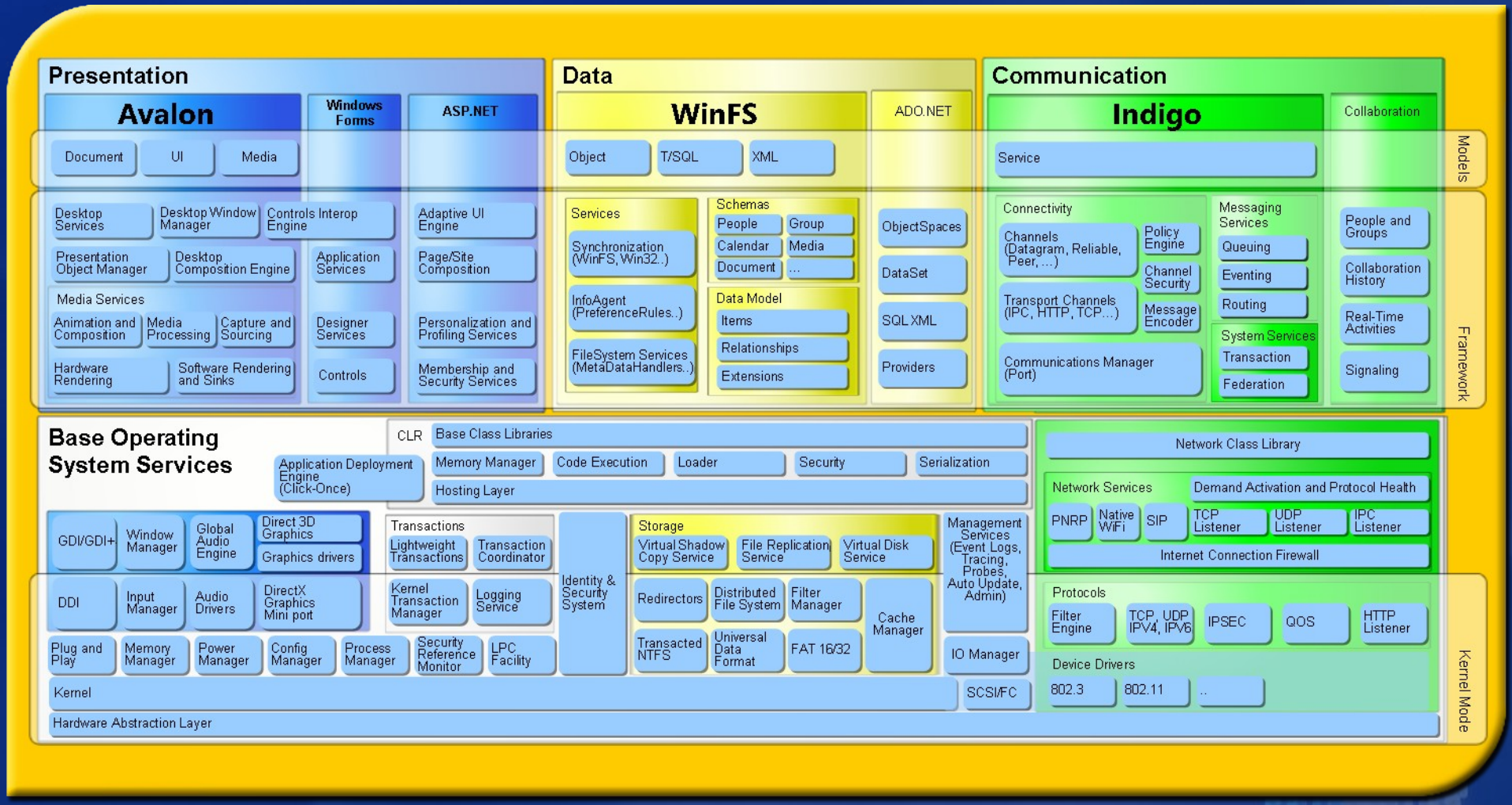
"Longhorn“: A New Approach

- Previously, the relationship between Microsoft Windows® internals (GDI, User) and tools (Microsoft Visual Basic®, ATL, MFC, Windows Forms) has been one of “layers”
  - Different technologies (C vs. Visual Basic, managed vs. unmanaged)
- In "Longhorn", the approach is to build a common stack, on managed code
  - The core of Windows is managed code
- This allows for significant developer benefits
  - Consistency across API sets
  - No “cliff” from wrappers or inconsistency based on customization

Longhorn Architecture



Longhorn Architecture

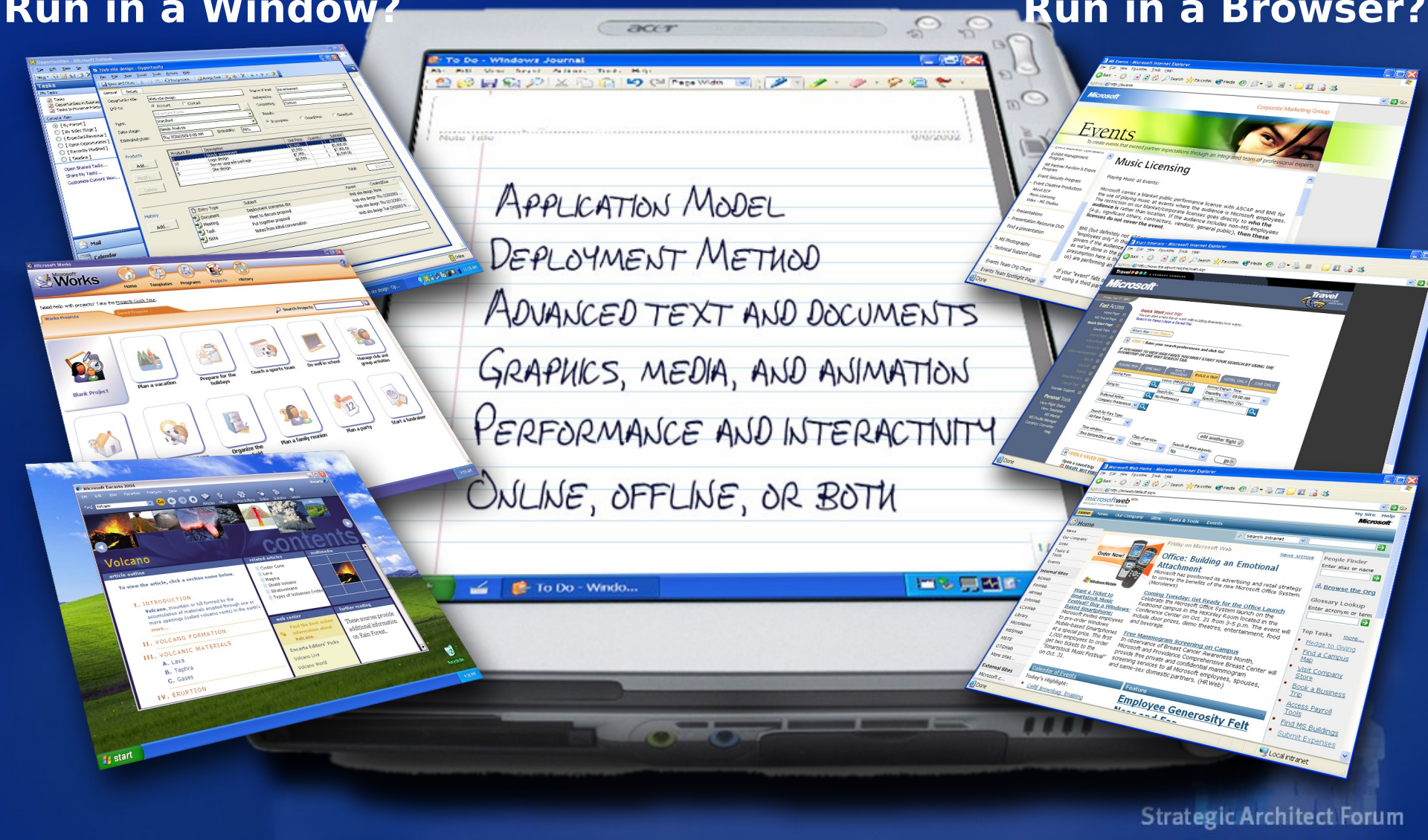


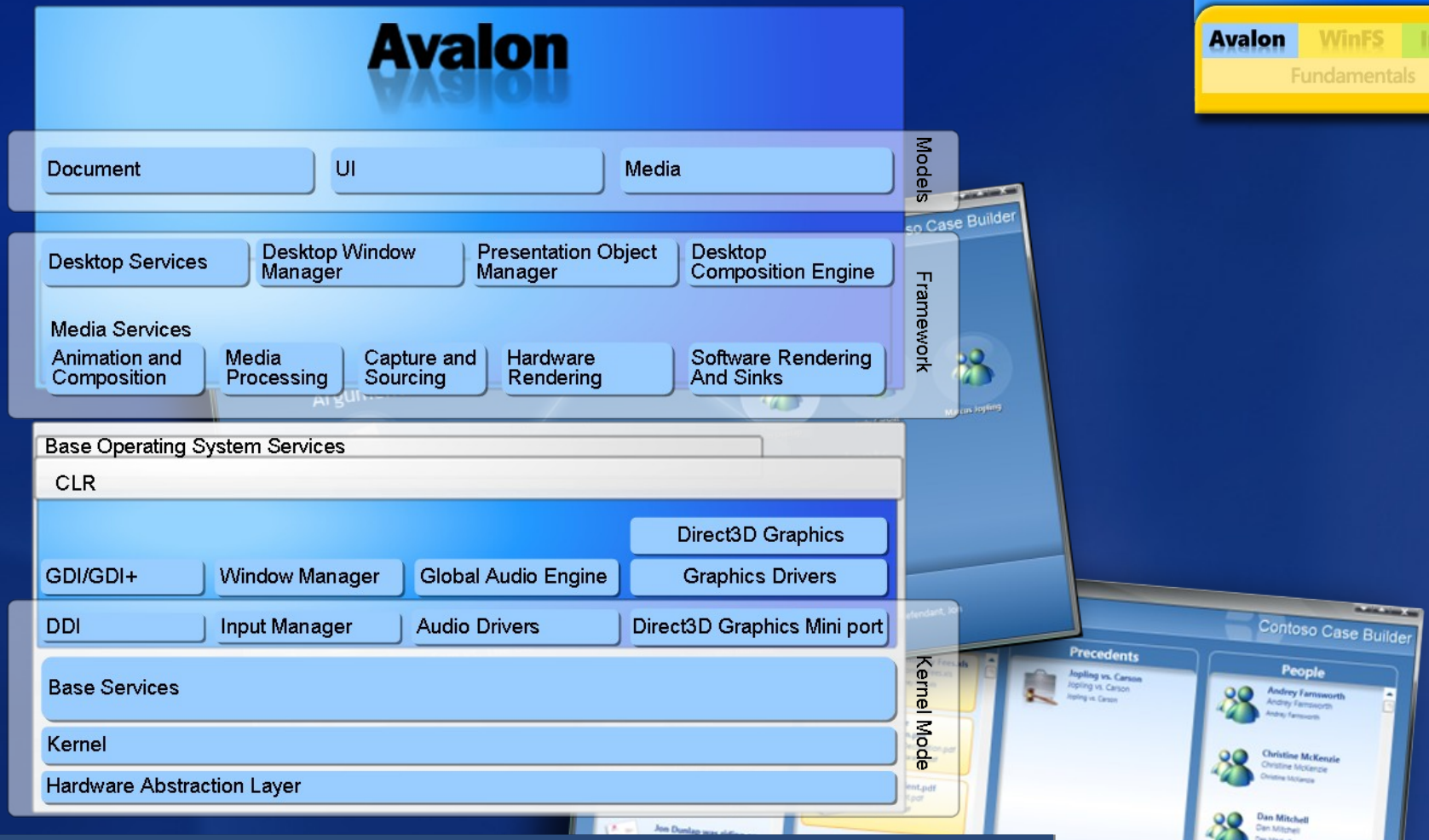
Presentation Challenges

WinFX
Avalon | WinFS | Indigo
Fundamentals

Run in a Window?

Run in a Browser?





- **Unified Presentation Model for Windows Applications, Web Applications, Graphics/Media/Animation**
- **Integrated, Vector-Based Compositing Engine**
- **Native Support for Advanced Input**

Declarative Programming for Windows

Code Named "XAML"

- **Markup for Windows**
  - **Build applications in simple declarative statements**
  - **Easy to learn, write, and read**
- **Code and Content Are Separate**
  - **Streamline collaboration between designers and developers**
- **Easy for Tools to Consume and Generate**
- **Control Extensibility and Composition**
  - **Customize or go beyond the default set of controls provided as part of the platform**

XAML

```
<Button Width="100px">
OK
  <Button.Background>
    LightBlue
  </Button.Background>
</Button>
```

C#

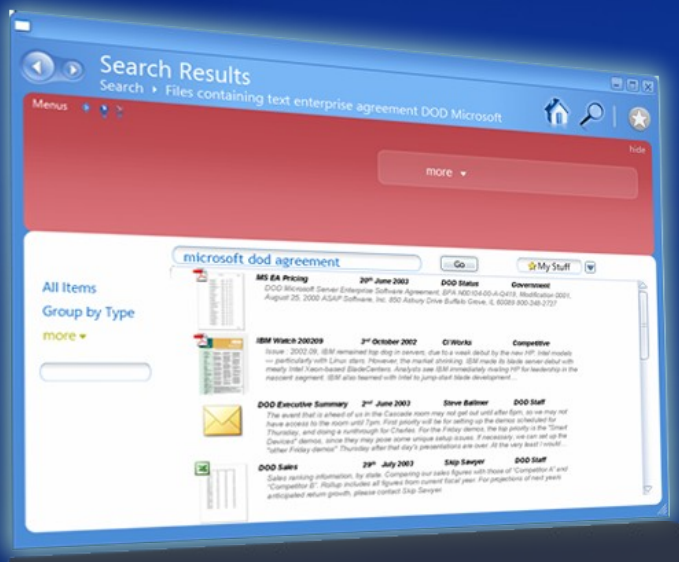
```
Button b1 = new Button();
b1.Content = "OK";
b1.Background = new
SolidColorBrush(Colors.LightBl
ue);
b1.Width = new Length(100);
```

VB.NET

```
Dim b1 As New Button
b1.Content = "OK"
b1.Background = New
SolidColorBrush(Colors.LightBl
ue)
b1.Width = New Length(100)
```

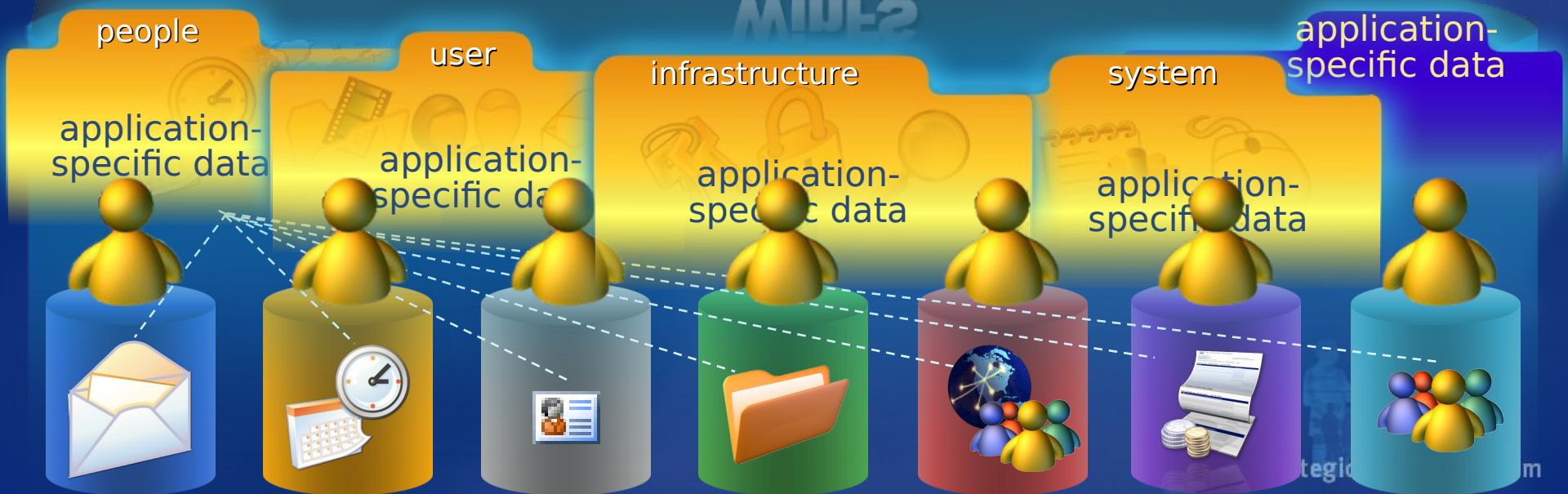
Storage Challenges

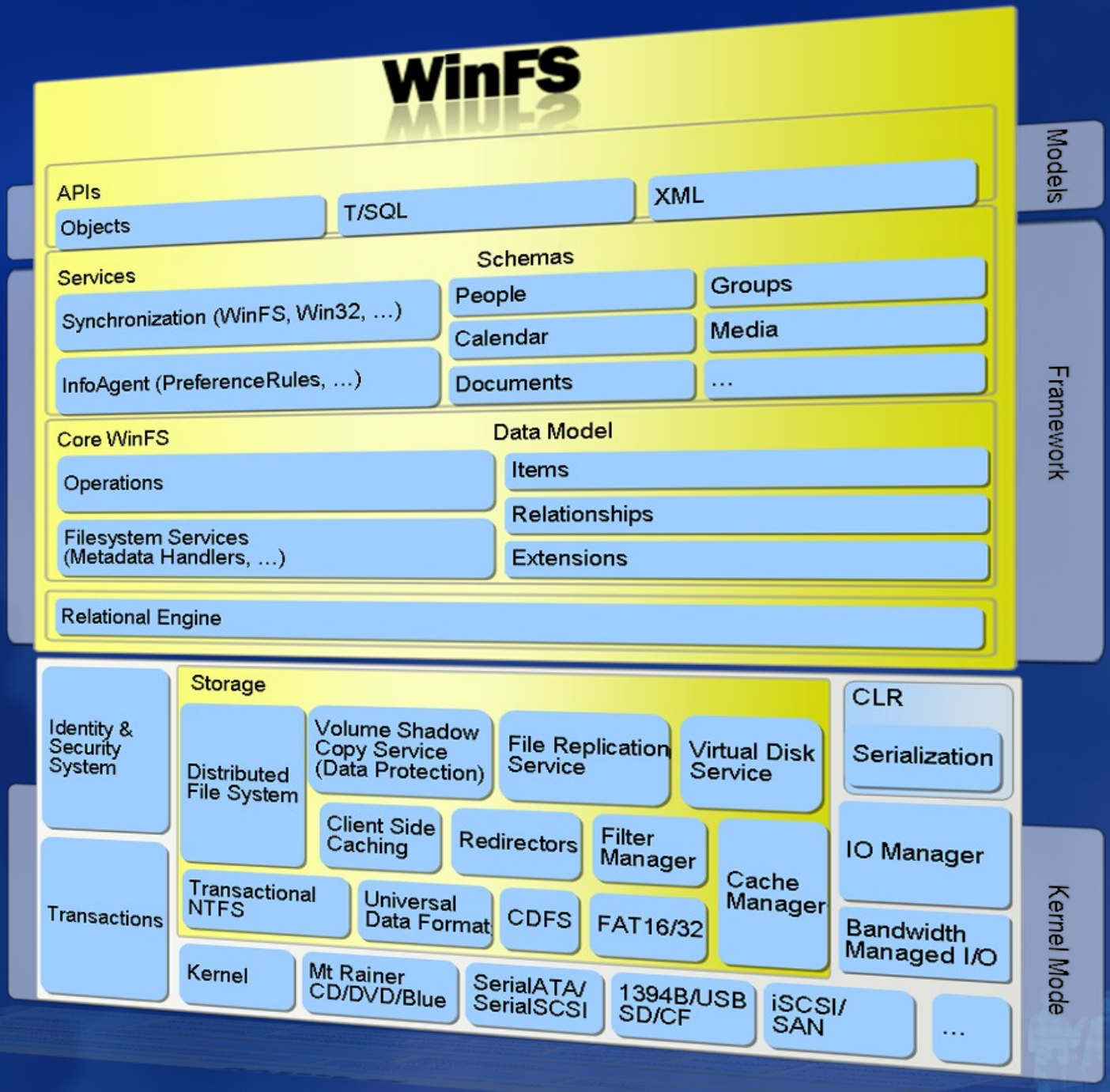
- Data Is Trapped Inside Application Silos
  - Each has its own schema and store
- Relationships Are Buried
  - Software doesn't surface the obvious links
- Shell Views Are Tied to Folder Hierarchy



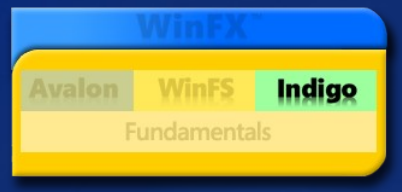
- Extensible XML Schemas
- Logical Views
- Programmatic Relationships
- Synchronization Service
- Information Agents

WinFS




WinFS
APIs
Objects
T/SQL
XML
Models
Services
Synchronization (WinFS, Win32, …)
InfoAgent (PreferenceRules, …)
Schemas
People
Groups
Calendar
Media
Documents
…
Framework
Core WinFS
Operations
Filesystem Services (Metadata Handlers, …)
Data Model
Items
Relationships
Extensions
Relational Engine
Identity & Security System
Transactions
Storage
Distributed File System
Volume Shadow Copy Service (Data Protection)
File Replication Service
Virtual Disk Service
Client Side Caching
Redirectors
Filter Manager
Cache Manager
Transactional NTFS
Universal Data Format
CDFS
FAT16/32
Kernel
Mt Rainer CD/DVD/Blue
SerialATA/ SerialSCSI
1394B/USB SD/CF
iSCSI/ SAN
…
CLR
Serialization
IO Manager
Bandwidth Managed I/O
Kernel Mode

Communication
Challenges



Web services are a great start but…

How do I make it work over peer-to-peer and enterprise networks?

How do I ensure reliable connections over networks that can fail?

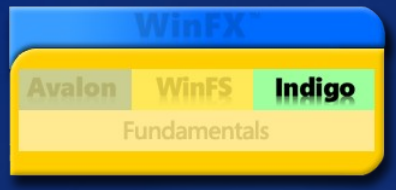
How do I secure my connections?

How do I create applications that cross trust boundaries?

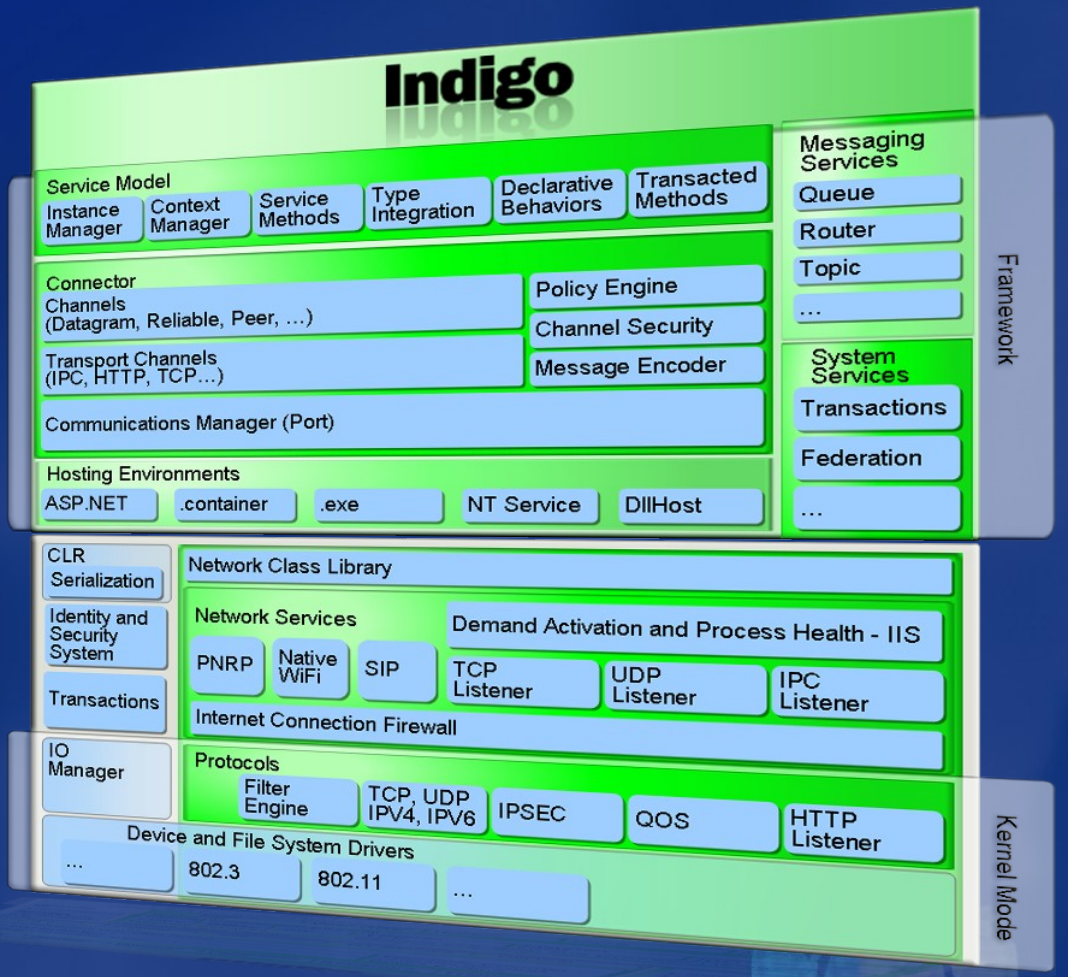
Should I program to objects or services?

Windows Communication

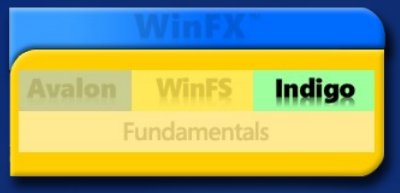
Code Named "Indigo"



- Advanced Web Services
  - Secure, reliable, transacted
  - Heterogeneous interoperability
- Powerful Messaging Capabilities
- Programming Model Extends Existing Capabilities
- Simplifies Building Services



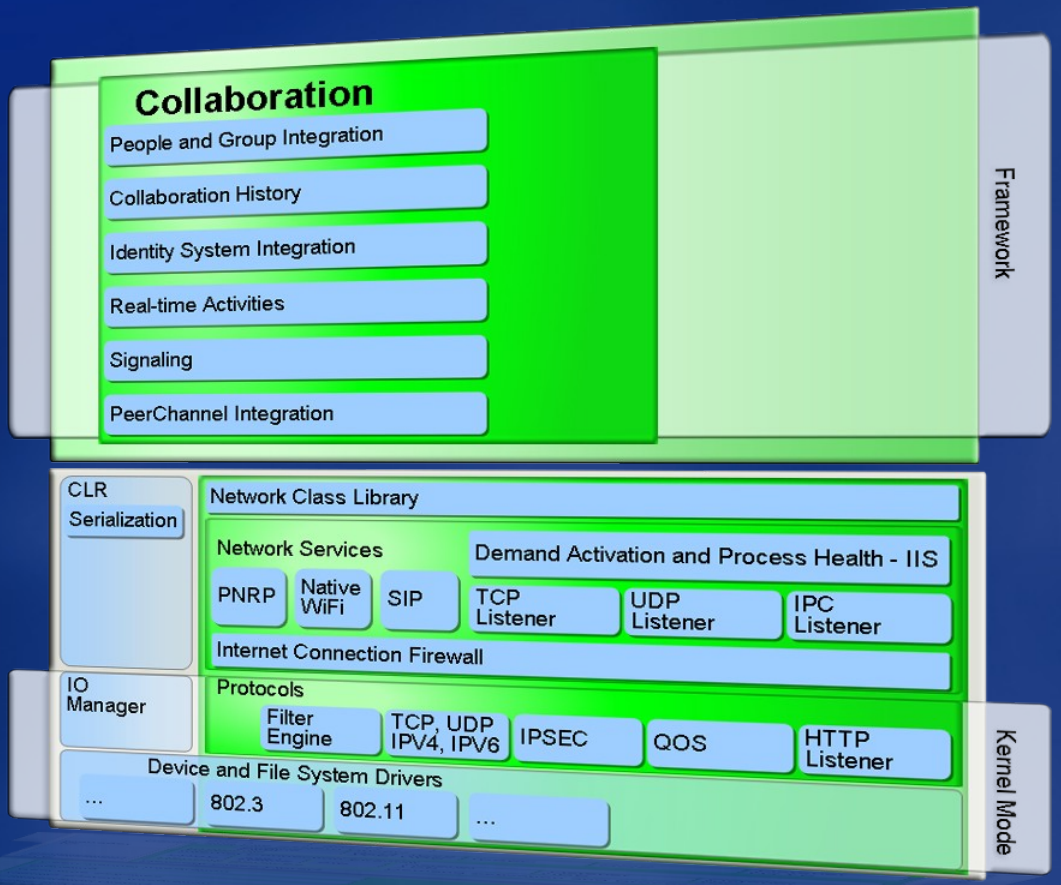
Collaboration
Challenges



- Too Many Ways to Manage Contacts
- No Single Real-Time Communications (RTC) API for IT-Managed and P2P Environments
- Current RTC APIs Are Monolithic and Low-Level
  - No UI controls for basics (text chat, video chat, etc.)
- Partitioned Networks Create Obstacles
  - No dynamic, consistent namespace
  - No standard security infrastructure

Windows Collaboration

- Single Source of Contacts *and Presence*
- Modular Platform with UI Controls for Chat, a/v, Data Collaboration, Telephony
- Single Platform for Enterprises and Peer-to-Peer
- Easy to Traverse Networks, Namespaces



The Road to Longhorn

Microsoft .net Framework

Move to managed code now to make it easy to fully exploit "Longhorn"

Win32 + WinFX™

Use some "Longhorn" features in your existing Microsoft Win32®-based applications

WinFX™

Take full advantage of Microsoft WinFX™ and the new application model

Win32

Existing applications continue to run on "Longhorn" as-is

Questions?

Microsoft®